



# **Hi3516CV500/Hi3516DV300/Hi3516AV300 SDK 安装及升级使用说明**

文档版本 01

发布日期 2019-09-12

**版权所有 © 上海海思技术有限公司 2019。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

#### **商标声明**



**HISILICON**、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

#### **注意**

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## **上海海思技术有限公司**

地址：                    深圳市龙岗区坂田华为总部办公楼                    邮编：518129

网址：                    <http://www.hisilicon.com/cn/>

客户服务邮箱：          [support@hisilicon.com](mailto:support@hisilicon.com)



# 前言

## 概述

本文为 Hi3516CV500/Hi3516DV300/Hi3516AV300 SDK 的安装及升级使用说明，方便使用者能快速在 Hi3516CV500/Hi3516DV300/Hi3516AV300 DEMB 板上搭建好 SDK 运行环境。下文以 Hi3516CV500 为例进行说明，如无特殊说明，Hi3516DV300/Hi3516AV300 与 Hi3516CV500 一致。

## 产品版本

与本文档相对应的产品版本如下。

| 产品名称    | 产品版本 |
|---------|------|
| Hi3516C | V500 |
| Hi3516D | V300 |
| Hi3516A | V300 |

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

### 文档版本 01 (2019-09-12)

2.1.2 小节涉及修改



## 文档版本 00B03 (2019-03-12)

第 3 次临时版本发布。

添加 Hi3516AV300 相关内容

## 文档版本 00B02 (2018-09-30)

第 2 次临时版本发布。

2.1.2 小节步骤 3 涉及修改

4.4 小节步骤 2 涉及修改

5.1、5.2 小节涉及修改

## 文档版本 00B01 (2018-09-06)

第 1 次临时版本发布。



## 目 录

|                                      |    |
|--------------------------------------|----|
| 前 言.....                             | i  |
| 1 首次安装 SDK.....                      | 1  |
| 1.1 Hi3516C V500 SDK 包位置 .....       | 1  |
| 1.2 解压缩 SDK 包.....                   | 1  |
| 1.3 展开 SDK 包内容.....                  | 1  |
| 1.4 在 linux 服务器上安装交叉编译器.....         | 1  |
| 1.5 编译 osdrv .....                   | 2  |
| 1.6 SDK 目录介绍.....                    | 2  |
| 2 安装、升级 Hi3516C V500 DEMO 板开发环境..... | 4  |
| 2.1 烧写 uboot、kernel、fs.....          | 4  |
| 2.1.1 准备工作 .....                     | 4  |
| 2.1.2 操作步骤 .....                     | 4  |
| 3 开发前环境准备.....                       | 8  |
| 3.1 管脚复用.....                        | 8  |
| 3.2 连接串口.....                        | 8  |
| 3.3 NFS 环境 .....                     | 8  |
| 4 使用 SDK 和 DEMO 板进行开发 .....          | 9  |
| 4.1 开启 Linux 下的网络 .....              | 9  |
| 4.2 使用 NFS 文件系统进行开发.....             | 9  |
| 4.3 开启 telnet 服务.....                | 9  |
| 4.4 运行 MPP 业务.....                   | 10 |
| 5 地址空间分配与使用 .....                    | 11 |
| 5.1 DDR 内存管理说明 .....                 | 11 |
| 5.2 DEMO 板 DDR 内存管理示意.....           | 11 |



# 1 首次安装 SDK

如果您已安装过 SDK，可以直接参看 [2 安装、升级 Hi3516C V500 DEMO 板开发环境](#)。

## 1.1 Hi3516C V500 SDK 包位置

在"Hi3516C V500\*\*\*/01.software/board"目录下，您可以看到一个 Hi3516C V500\_SDK\_Vx.x.x.x.tgz 的文件，该文件就是 Hi3516C V500 的软件开发包。

## 1.2 解压缩 SDK 包

在 linux 服务器上（或者一台装有 linux 的 PC 上，主流的 linux 发行版本均可以），使用命令：`tar -zxf Hi3516CV500_SDK_Vx.x.x.x.tgz`，解压缩该文件，可以得到一个 Hi3516CV500\_SDK\_Vx.x.x.x 目录。

## 1.3 展开 SDK 包内容

返回 Hi3516CV500\_SDK\_Vx.x.x.x 目录，运行 `./sdk.unpack`（请用 root 或 sudo 权限执行）将会展开 SDK 包打包压缩存放的内容，请按照提示完成操作。

如果您需要通过 WINDOWS 操作系统中转拷贝 SDK 包，请先运行 `./sdk.cleanup`，收起 SDK 包的内容，拷贝到新的目录后再展开。

## 1.4 在 linux 服务器上安装交叉编译器

在发布包 Hi3516C V500R001C01SPCxxx.rar 所在的目录中下载工具链文件。

**注意：**安装交叉编译器需要有 `sudo` 权限或者 `root` 权限。

1) 安装 himix200 交叉编译器：



解压 `tar -xzf arm-himix200-linux.tgz`, 运行 `chmod +x arm-himix200-linux.install`, 然后运行 `./arm-himix200-linux.install` 即可。

2) 执行 `source /etc/profile`, 安装交叉编译器的脚本配置的环境变量就可以生效了, 或者请重新登陆也可。

## 1.5 编译 osdrv

参见 `osdrv` 目录下 `readme`

## 1.6 SDK 目录介绍

Hi3516C V500\_SDK\_Vx.x.x.x 目录结构如下:

```
-- smp                                #smp 目录
|--a7_linux
    |-- drv                            # drv 目录
    |   |-- extdrv                      # 板级外围驱动源代码
    |   |-- interdrv                    # mipi,cipher 等驱动源代码
    |-- mpp                            # 存放单核媒体处理平台的目录
    |   |-- component                  # mpp 组件
    |   |   |-- isp                    # isp 相关组件
    |   |-- init                       # 内核模块的初始化源代码
    |   |-- obj                        # 内核模块的 obj 文件
    |   |-- include                    # 头文件
    |   |-- ko                         # 内核 ko 模块
    |   |-- lib                        # 用户态 lib 库
    |   |-- sample                     # 样例源代码
    |   |-- tools                      # 媒体处理相关工具
    |   |-- cfg.mak                    # mpp 配置文件
    |   |-- Makefile.param             # mpp 全局编译选项
    |   |-- Makefile.linux.param       # mpp linux 编译选项
    |-- osal                           # 存放操作系统适配层的头文件和源文件的目录
    |   |-- include                    # 存放操作系统适配层的头文件的目录
    |   |-- linux                      # 存放 linux 系统适配层的源文件的目录
```



```
|-- osdrv                                # 存放操作系统及相关驱动的目录
    |-- component                        # 组件源代码
    |-- opensource                       # opensource 源代码
    |   |-- busybox                     # busybox 源代码
    |   |-- kernel                      # linux 内核源代码
    |   |-- uboot                       # uboot 源代码
    |-- platform                         # 平台文件
    |-- pub                             # 编译好的镜像、工具、drv 驱动等
    |-- tools                           # 工具源代码
    |-- readme_cn.txt                   # osdrv 中文使用说明
    |-- readme_en.txt                   # osdrv 英文使用说明
    |-- .....                           #
    |-- Makefile                        # osdrv Makefile
|-- package                             # 存放 SDK 各种压缩包的目录
    |-- drv.tgz                         # drv 压缩包
    |-- mpp_smp_linux.tgz               # 媒体处理平台软件压缩包
    |-- osal.tgz                        # 操作系统适配层源码压缩包
    |-- osdrv.tgz                       # linux 内核/uboot/rootfs/tools 源码压缩包
|-- scripts                             # 存放 shell 脚本的目录
|-- sdk.cleanup                         # SDK 清理脚本
|-- sdk.unpack                          # SDK 展开脚本
```





# 2 安装、升级 Hi3516C V500 DEMO 板开发环境

本章节以 Hi3516C V500 DEMO 板为例，介绍烧写 u-boot、内核以及文件系统的方法。

## 2.1 烧写 u-boot、kernel、fs

### 2.1.1 准备工作

首先，请阅读文档《Hi3516C V500 Demo 单板用户指南》，了解 Hi3516C V500 DEMO 板硬件的功能、结构、接口等信息。

1. 如果您拿到的单板没有 u-boot，就需要使用 HiTool 工具进行烧写。HiTool 工具位置放在 Hi3516C V500\*\*\*/01.software/pc/HiTool，使用说明请参见 ReleaseDoc\zh\01.software\pc\HiTool 下的《HiBurn 工具使用指南》。
2. 如果您拿到的单板中已经有 u-boot，可以按照以下步骤使用网口烧写 u-boot、kernel 及 rootfs 到 Flash 中。

本章所有的烧写操作都是在串口上进行，烧写到 SPI NOR Flash 上。

### 2.1.2 操作步骤

#### 步骤 1 配置 tftp 服务器

可以使用任意的 tftp 服务器，将 package/smp\_image\_uclibc\_xxx(或 image\_uclibc\_xxx)下的相关文件拷贝到 tftp 服务器目录下。

#### 步骤 2 参数配置

单板上电后，敲任意键进入 u-boot。设置 serverip（即 tftp 服务器的 ip）、ipaddr（单板 ip）和 ethaddr（单板的 MAC 地址）。

```
setenv serverip xx.xx.xx.xx
setenv ipaddr xx.xx.xx.xx
setenv ethaddr xx:xx:xx:xx:xx:xx
```



```
setenv netmask xx.xx.xx.xx
```

```
setenv gatewayip xx.xx.xx.xx
```

ping serverip, 确保网络畅通。

### 步骤 3 SMP 版本烧写映像文件到 SPI NOR Flash

地址空间说明

|       |        |        |  |
|-------|--------|--------|--|
| 1M    | 4M     | 11M    |  |
| ----- | -----  | -----  |  |
| boot  | kernel | rootfs |  |

以下的操作均基于图示的地址空间分配, 您也可以根据实际情况进行调整。

#### 1. 烧写 u-boot

```
mw.b 82000000 0xff 80000
tftp 82000000 u-boot-hi3516cv500.bin
sf probe 0;sf erase 0 80000;sf write 82000000 0 80000
```

#### 2. 烧写内核

```
mw.b 82000000 0xff 400000
tftp 82000000 uImage_hi3516cv500_smp
sf probe 0;sf erase 100000 400000;sf write 82000000 100000 400000
```

#### 3. 烧写文件系统

```
mw.b 82000000 0xff b00000
tftp 82000000 rootfs_hi3516cv500_64k.jffs2
sf probe 0;sf erase 500000 b00000;sf write 82000000 500000 b00000
```

#### 4. 设置启动参数 (注意 linux-4.9.y kernel 默认文件系统只读, 需要在 bootargs 中加入 rw 选项, 文件系统才可读写)

```
setenv bootargs 'mem=64M console=ttyAMA0,115200 root=/dev/mtdblock2
rootfstype=jffs2 rw mtdparts=hi_sfc:1M(boot),4M(kernel),11M(rootfs)'
setenv bootcmd 'sf probe 0;sf read 0x82000000 0x100000 0x400000;bootm 0x82000000'
saveenv
```

#### 5. 重启系统

```
reset
```

### 步骤 4 SMP 版本烧写映像文件到 SPI nand Flash

以 64MB SPI nand Flash 为例, 其中步骤 4、5 仅在 yaff2 文件系统时参考, 步骤 6、7 仅在 UBI 文件系统时参考。

#### 1. 地址空间说明

|       |       |       |  |
|-------|-------|-------|--|
| 1MB   | 4MB   | 32MB  |  |
| ----- | ----- | ----- |  |



| boot | kernel | rootfs |

以下操作均基于图示的地址空间分配，您也可以根据实际情况进行调整。

2. 烧写 u-boot

```
mw.b 82000000 0xff 80000
tftp 82000000 u-boot-hi3516cv500.bin
nand erase 0x0 0x80000
nand write 0x82000000 0x0 0x80000
```

3. 烧写内核

```
mw.b 82000000 0xff 400000
tftp 82000000 uImage_hi3516cv500_smp
nand erase 0x100000 0x400000
nand write 0x82000000 0x100000 0x400000
```

4. 烧写文件系统

```
mw.b 82000000 0xff 1000000
tftp 82000000 rootfs_hi3516cv500_4k_24bit.yaffs2
nand erase 0x500000 0x2000000
nand write.yaffs 0x82000000 0x500000 0xd06398 #注意：d06398 为 rootfs 文件
实际大小（16 进制）
```

5. 设置启动参数（注意 linux-4.9.y kernel 默认文件系统只读，需要在 bootargs 中加入 rw 选项，文件系统才可读写）

```
setenv bootargs 'mem=64M console=ttyAMA0,115200 root=/dev/mtdblock2
rootfstype=yaffs2 rw mtdparts=hinand:1M(boot),4M(kernel),32M(rootfs)'
setenv bootcmd 'nand read 0x82000000 0x100000 0x400000;bootm 0x82000000'
saveenv
```

6. 烧写 ubifs 文件系统

```
mw.b 0x82000000 0xff 0x3200000
tftp 0x82000000 rootfs_hi3516cv500_4k_256k_50M.ubifs
nand erase 0x500000 0x3200000
nand write 0x82000000 0x500000 0x3200000
```

7. 设置启动参数（注意 linux-4.9.y kernel 默认文件系统只读，需要在 bootargs 中加入 rw 选项，文件系统才可读写）

```
setenv bootargs 'mem=64M console=ttyAMA0,115200 ubi.mtd=2 root=ubi0:ubifs
rootfstype=ubifs rw mtdparts=hinand:1M(boot),4M(kernel),50M(rootfs.ubifs)'
setenv bootcmd 'nand read 0x82000000 0x100000 0x400000;bootm 0x82000000'
saveenv
```

8. 重启系统

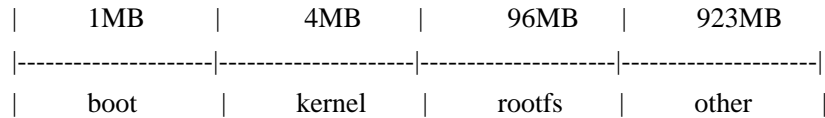
```
reset
```



## 步骤 5 烧写映像文件到 EMMC。

以 1024MB EMMC 为例：

### 1. 地址空间说明



以下操作均基于图示的地址空间分配，您也可以根据实际情况进行调整。

### 2. 烧写 u-boot

```
mw.b 0x82000000 0xff 0x80000
tftp 0x82000000 u-boot-hi3516cv500.bin
mmc write 0x0 0x82000000 0x0 0x400
```

### 3. 烧写内核

```
mw.b 0x82000000 0xff 0x400000
tftp 0x82000000 uImage_hi3516cv500_smp
mmc write 0 0x82000000 0x800 0x2000
```

### 4. 烧写文件系统

```
mw.b 0x82000000 0xff 0x6000000
tftp 0x82000000 rootfs_hi3516cv500_96M.ext4
mmc write.ext4sp 0 0x82000000 0x2800 0x30000
```

### 5. 设置启动参数（注意 linux-4.9.y kernel 默认文件系统只读，需要在 bootargs 中加入 rw 选项，文件系统才可读写）

```
setenv bootargs 'mem=64M console=ttyAMA0,115200 root=/dev/mmcblk0p3 rw
rootfstype=ext4 rootwait blkdevparts=mmcblk0:1M(boot),4M(kernel),96M(rootfs)'
setenv bootcmd 'mmc read 0x0 0x82000000 0x800 0x2000;bootm 0x82000000'
saveenv
```

### 6. 重启系统

```
reset
```

----结束



# 3 开发前环境准备

---

## 3.1 管脚复用

无

## 3.2 连接串口

通过 DEMO 板的串口连接到 CPU。

## 3.3 NFS 环境

通过 DEMO 板的网口连接 NFS。



# 4 使用 SDK 和 DEMO 板进行开发

## 4.1 开启 Linux 下的网络

步骤 1 设置网络

```
ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx;  
ifconfig eth0 xx.xx.xx.xx netmask xx.xx.xx.xx;  
route add default gw xx.xx.xx.xx
```

步骤 2 然后 ping 一下其他机器，如无意外，网络将能正常工作。

----结束

## 4.2 使用 NFS 文件系统进行开发

步骤 1 在开发阶段，推荐使用 NFS 作为开发环境，可以省去重新制作和烧写根文件系统的工作。

步骤 2 挂载 NFS 文件系统的操作命令：

```
mount -t nfs -o nolock -o tcp -o rsize=32768,wsiz=32768 xx.xx.xx.xx:/your-nfs-path /mnt
```

步骤 3 然后就可以在/mnt 目录下访问服务器上的文件，并进行开发工作。

----结束

## 4.3 开启 telnet 服务

网络正常后，运行命令 `telnetd&` 就可以启动单板 telnet 服务，然后才能使用 telnet 登录到单板。



## 4.4 运行 MPP 业务

步骤 1 在串口上，进入 mpp /ko 目录，加载驱动，例：

```
cd mpp/ko  
./load3516cv500 -i -sensor0 imx327
```

步骤 2 进入各 sample 目录下执行相应样例程序(sample 需要先在服务器上成功编译过)

```
cd mpp/sample /vio/smp  
./sample_vio 0
```

----结束



# 5 地址空间分配与使用

## 5.1 DDR 内存管理说明

- 对于 SMP 版本，DDR 内存分成两部分：
  - 一部分由 Linux 操作系统管理，称为 OS 内存。
  - 一部分由 osal 模块管理，供媒体业务单独使用，称为 MMZ 内存。
- 对于 AMP 版本，DDR 内存分成五部分：
  - 一部分由 IPCM 模块管理，称为 IPCM 内存。
  - 一部分由 Linux 操作系统管理，称为 Linux OS 内存
  - 一部分由 Linux 侧 osal 模块管理，供媒体业务单独使用，称为 Linux 侧 MMZ 内存。
  - 一部分由 Huawei LiteOS 操作系统管理，称为 Liteos OS 内存
  - 一部分由 Huawei LiteOS 侧 osal 模块管理，供媒体业务单独使用，称为 Liteos 侧 MMZ 内存。
- 对于 SMP 版本，OS 内存起始地址为 0x80000000，内存大小可通过 bootargs 进行配置，例如 [2.1.2 操作步骤](#)步骤 3 中的 setenv bootargs 'mem=64M ...'，表示分配给 Linux OS 操作系统内存为 64M，您可以根据实际情况进行调整。MMZ 内存由 osal 内核模块管理（smp/a7\_linux/mpp/ko/hi\_osal.ko），加载 osal 模块时，通过模块参数指定其起始地址及大小，可在 load 脚本中修改 MMZ 的起始地址 mmz\_start 及大小 mmz\_size。
- 对于 AMP 版本，内存的配置请参考 [5.2 DEMO 板 DDR 内存管理示意](#)中 AMP 版本的 DDR 配置说明。
- 请注意任何区域的内存划分都不能重叠。

## 5.2 DEMO 板 DDR 内存管理示意

以容量为 256M Bytes 的 DDR 内存为例，以下为根据本文档和 SDK 默认配置得到的内存管理示意图：





## SMP 版本

DDR:

```

|-----|-----| 0x80000000 # Memory managed by OS.
| 64MB   | Linux OS   |
|-----|-----| 0x84000000 # Memory managed by MMZ.
| 192M B | MMZ           |
|-----|-----| 0x90000000

```

注意:

- 用户在配置启动参数时需要设置 OS 的管理内存，“setenv bootargs 'mem=64M ...’”。
  - 对于 Hi3516CV500，建议设置为 64MB。
  - 对于 Hi3516DV300/Hi3516AV300，建议设置为 128MB。
 用户可以根据实际使用情况自行修改。
- load3516cv500 脚本(mpp/ko 目录下)默认的 MMZ 管理的内存地址从 0x84000000 起始，大小为 192MB，用户可以根据实际使用情况自行修改。
- load3516dv300/load3516av300 脚本(mpp/ko 目录下)默认的 MMZ 管理的内存地址从 0x88000000 起始，大小为 384MB，用户可以根据实际使用情况自行修改。
- 任何用途的内存区域地址空间都不能重叠。
- 如果有特殊应用，可以自行修改 load3516cv500/load3516dv300/load3516av300 脚本，进行 mmz 区域划分，如 “insmod hi\_osal.ko mmz\_allocator=hisi mmz=anonymous,0,0x84000000,32M:jpeg,0,0x86000000,2M”。

## AMP 版本

DDR:

```

|-----|-----| 0x80000000 # Memory managed by IPCM.
| 2MB    | IPCM       |
|-----|-----| 0x80200000 # Memory managed by Liteos OS.
| 30MB   | Liteos OS  |
|-----|-----| 0x82000000 # Memory managed by Linux OS.
| 96MB   | Linux OS   |
|-----|-----| 0x88000000 # Memory managed by Liteos MMZ.
| 336MB  | MMZ        |
|-----|-----| 0x9d000000 # Memory managed by Linux MMZ.
| 48MB   | MMZ        |
|-----|-----| 0xa0000000

```

注意:



- Linux 侧的 OS 内存在配置启动参数时设置 “setenv bootargs 'mem=96M ...’”。MMZ 内存通过 mpp/ko 目录下的 load3516cv500/load3516dv300/load3516av300 脚本配置，配置方式和 SMP 版本的 MMZ 内存配置方式一样。
- Huawei LiteOS 侧的 OS 内存和 MMZ 内存存在 osdrv/liteos/platform/bsp/board/hi3516cv500/include/board.h 中配置。其中：
  - DDR\_MEM\_ADDR: 物理内存的起始地址，默认值为 0x80000000。
  - DDR\_MEM\_SIZE: 物理内存的总大小，客户请根据实际的大小配置。
  - SYS\_MEM\_BASE: Huawei LiteOS OS 内存的起始地址。默认值为 0x80000000 加上 IPCM 管理的内存。
  - SYS\_MEM\_SIZE\_DEFAULT: Huawei LiteOS OS 内存的大小。
  - MMZ\_MEM\_BASE: Huawei LiteOS 侧 MMZ 内存的起始地址。
  - MMZ\_MEM\_LEN: Huawei LiteOS 侧 MMZ 内存的大小。

Huawei LiteOS 侧的 MMZ 内存还需要在 mpp/out/amp/a7\_liteos/init 目录下的 sdk\_int.c 文件中进行配置：

```
static unsigned long long mmz_start = 0x88000000;
```

```
static unsigned int mmz_size = 336;    //M Byte
```

以上两个变量分别对应 MMZ 的起始地址和大小，请确保和 board.h 中相应的变量保持一致。

- 任何用途的内存区域地址空间都不能重叠。